INDEX

EXP NO : 01	AZURE DEVOPS ENVIRONMENT SETUP

AIM

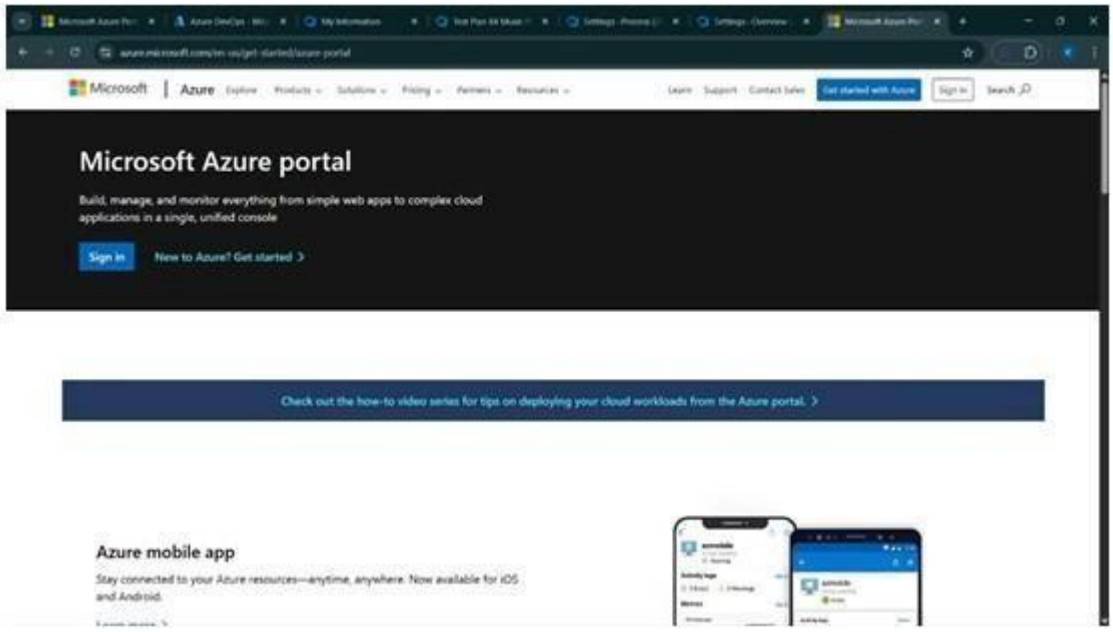
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

INSTALLATION

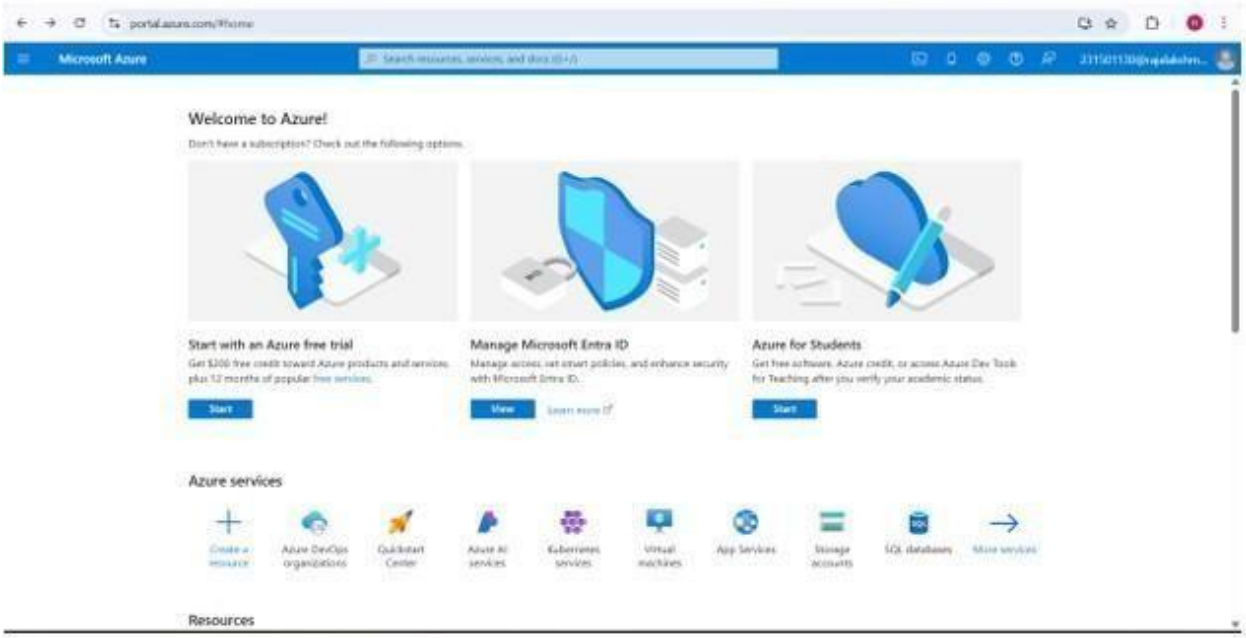
1.Open your web browser and go to the Azure website: https://azure.microsoft.com/en-us/getstarted/azure portal.

Sign in using your Microsoft account credentials.

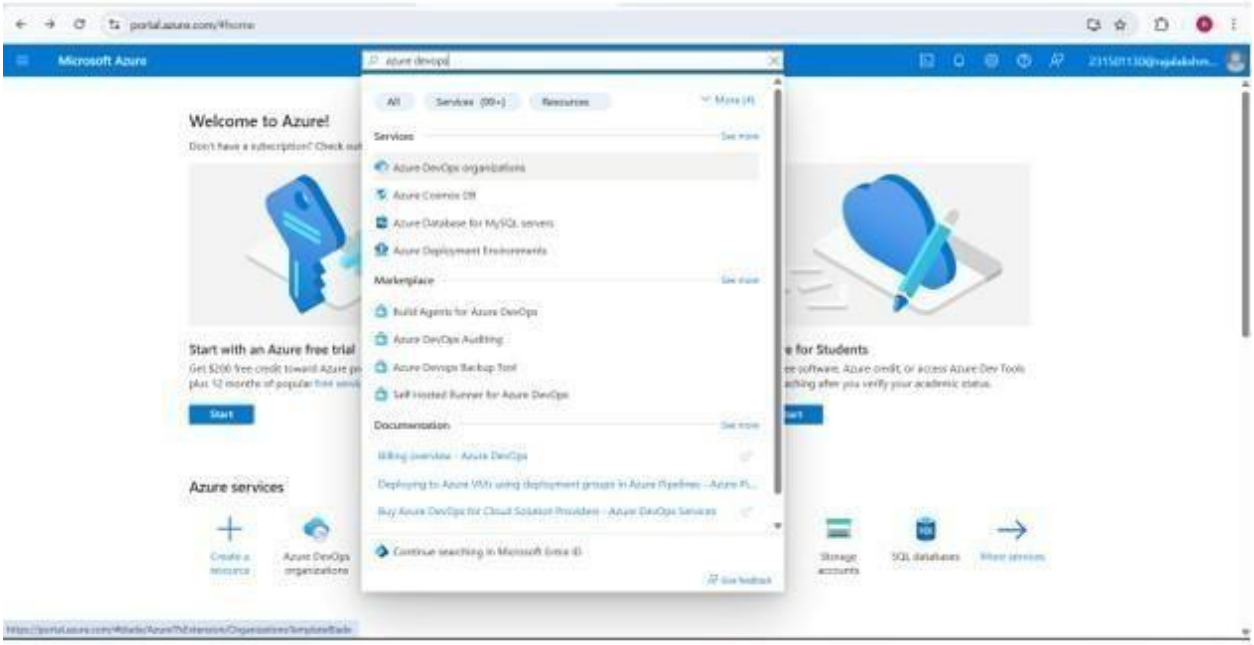
If you don't have a Microsoft account, you can create one here: https://signup.live.com/?lic=1



2.Azure home page



3.Open DevOps environment in the Azure platform by typing **Azure DevOps Organizations** in the search bar.



4.Click on the **My Azure DevOps Organization** link and create an organization and you should be taken to the Azure DevOps Organization Home page.

RESULT :

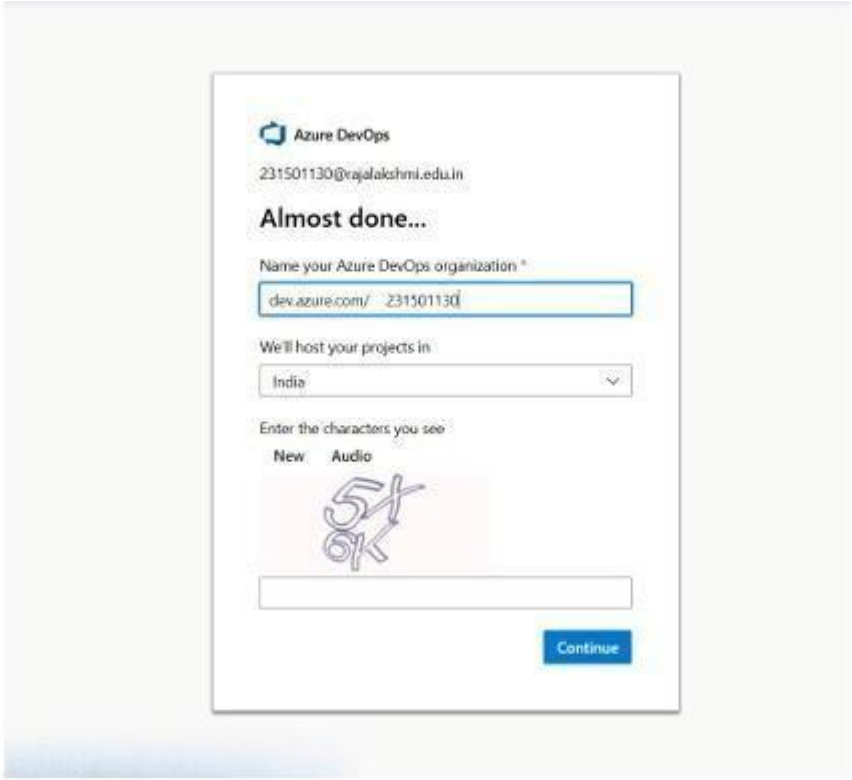
Successfully accessed the Azure DevOps environment and created a new organization through the Azure portal.

EXP NO:02	AZURE DEVOPS PROJECT SETUP AND USER STORY MANAGEMENT

AIM

To set up an Azure DevOps project for efficient collaboration and agile work management.

1.Create An Azure Account



2.Create the First Project in Your Organization

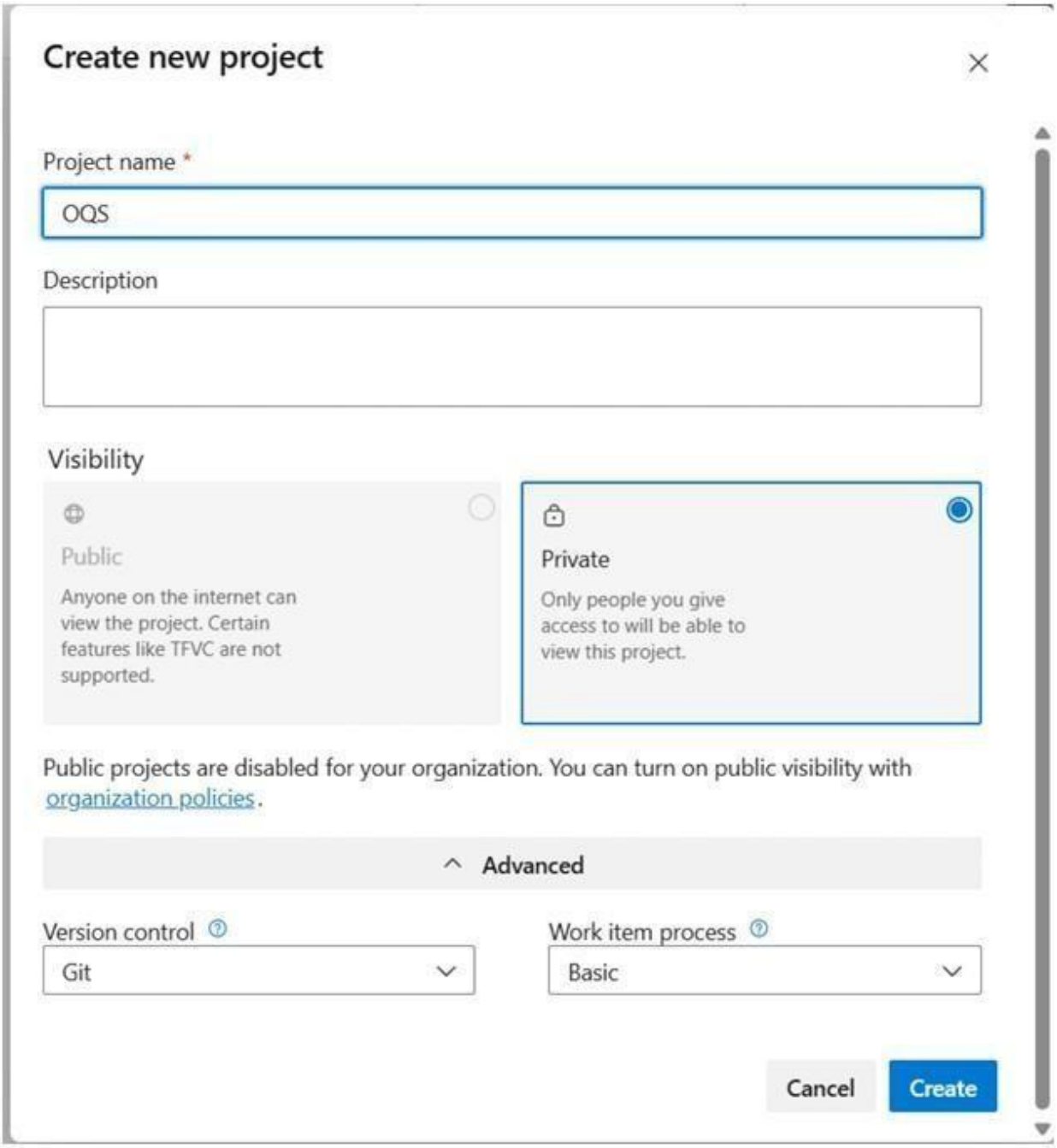
a. After the organization is set up, you'll need to create your first project. This is where you'll begin to manage code, pipelines, work items, and more.

b. On the organization's Home page, click on the New Project button.

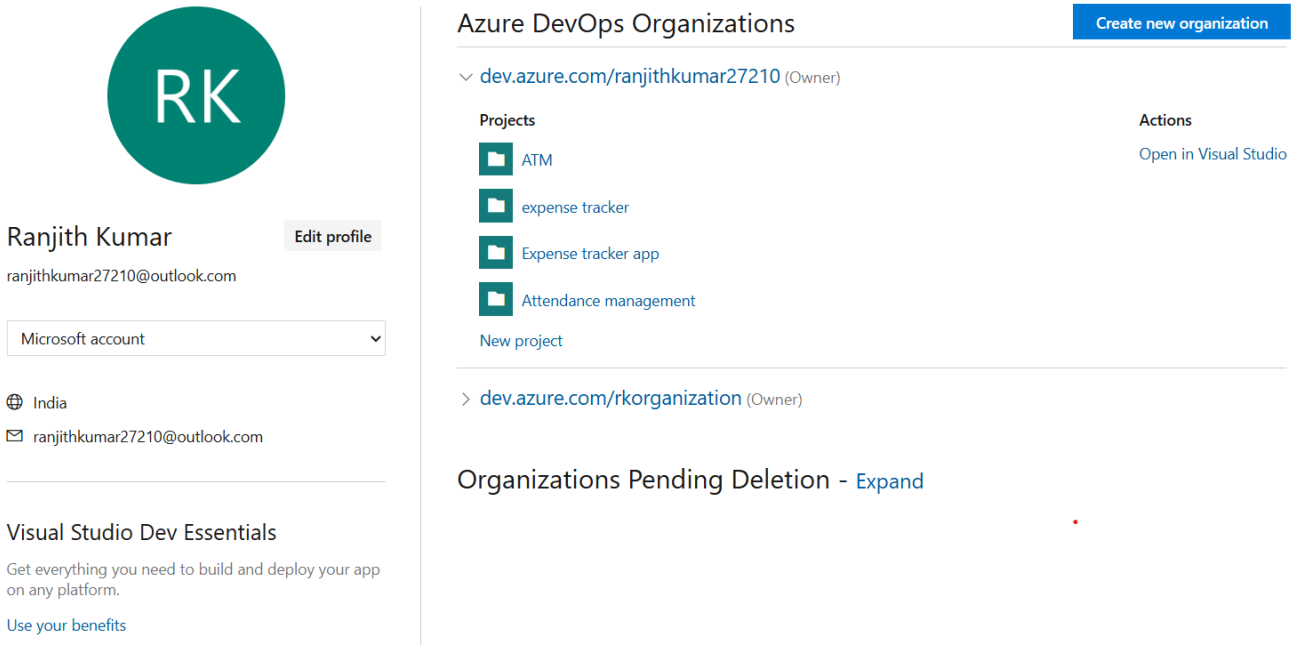
c. Enter the project name, description, and visibility options:

Name: Choose a name for the project (e.g., LMS).
Description: Optionally, add a description to provide more context about the project.
Visibility: Choose whether you want the project to be Private (accessible only to those invited) or Public (accessible to anyone).

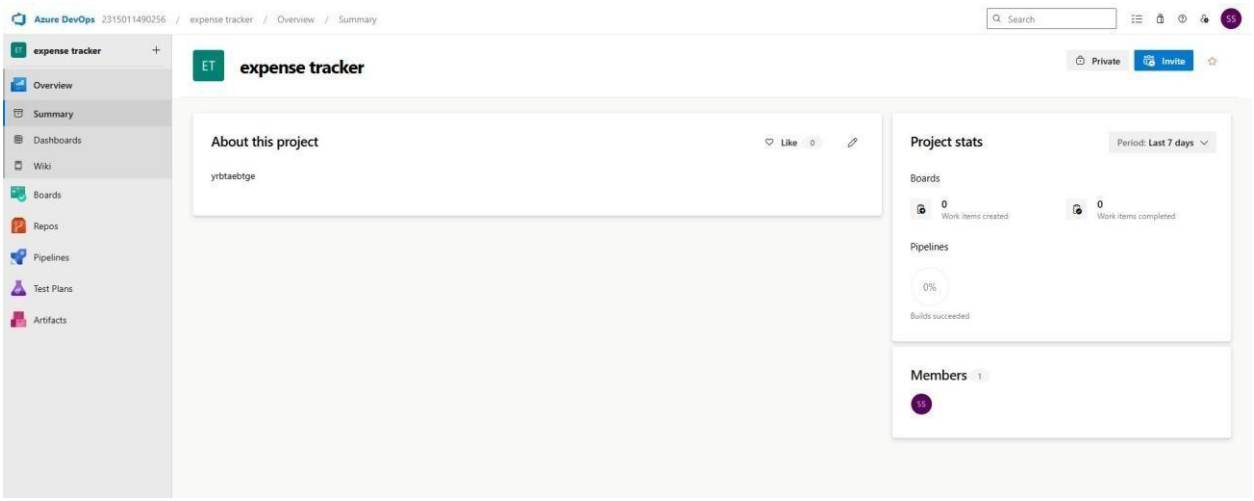
d. Once you've filled out the details, click Create to set up your first project.



3.Once logged in, ensure you are in the correct organization. If you're part of multiple organizations, you can switch between them from the top left corner (next to your user profile). Click on the Organization name, and you should be taken to the Azure DevOps Organization Home page.



4.Project dashboard

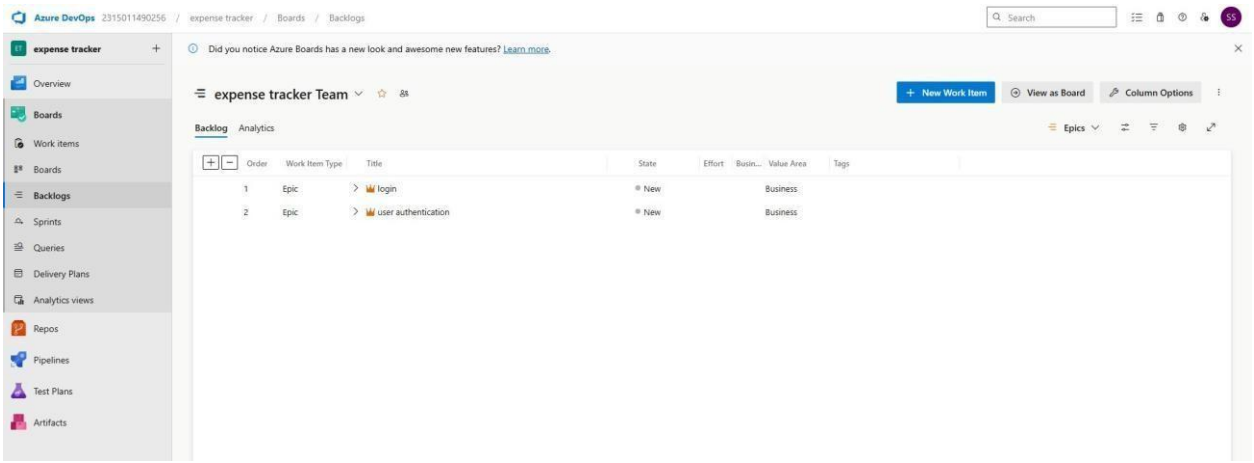


5.To manage user stories:

a. From the left-hand navigation menu, click on Boards. This will take you to the main Boards page, where you can manage work items, backlogs, and sprints.

b. On the work items page, you'll see the option to Add a work item at the top. Alternatively, you can find a + button or Add New Work Item depending on the view you're in. From the Add a work item dropdown, select User Story. This will open a form to enter details for the new User Story.

c.



RESULT :

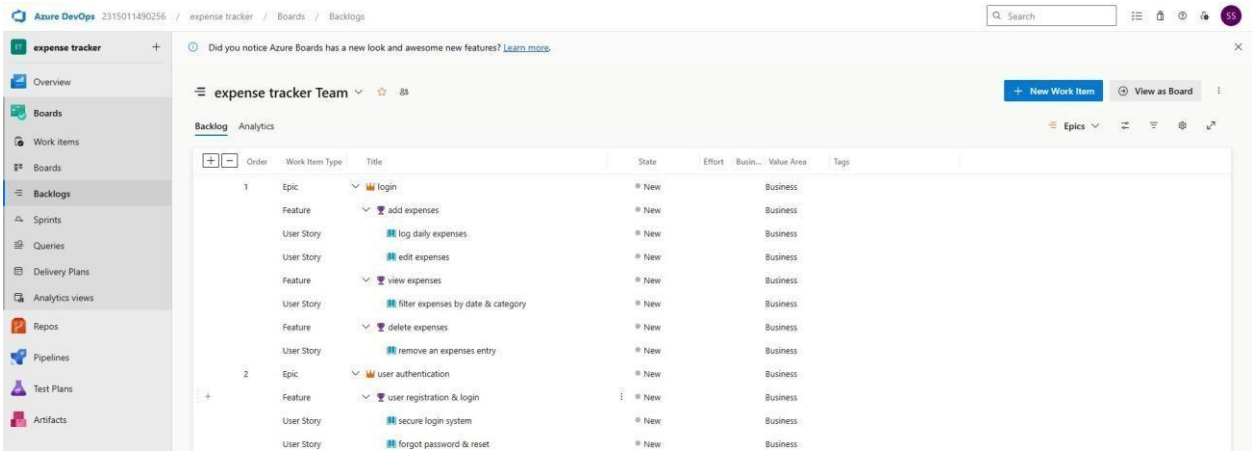
Successfully created an Azure DevOps project with user story management and agile workflow setup.

EXP NO:03	SETTING UP EPICS, FEATURES, AND USER STORIES FOR PROJECT PLANNING

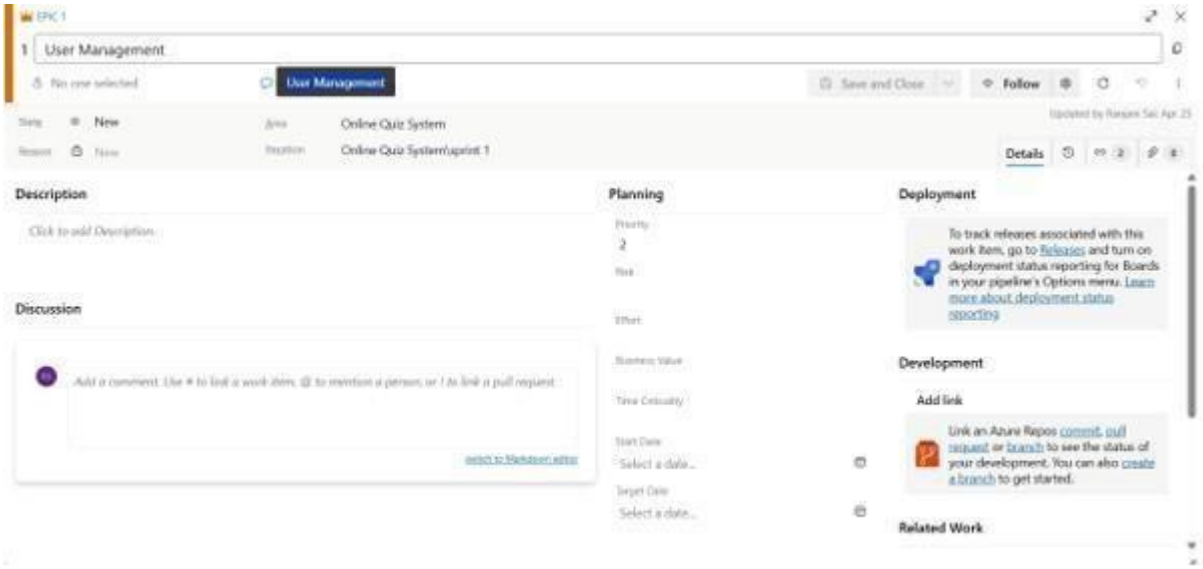
AIM

To learn about how to create epics, user story, features, backlogs for your assigned project

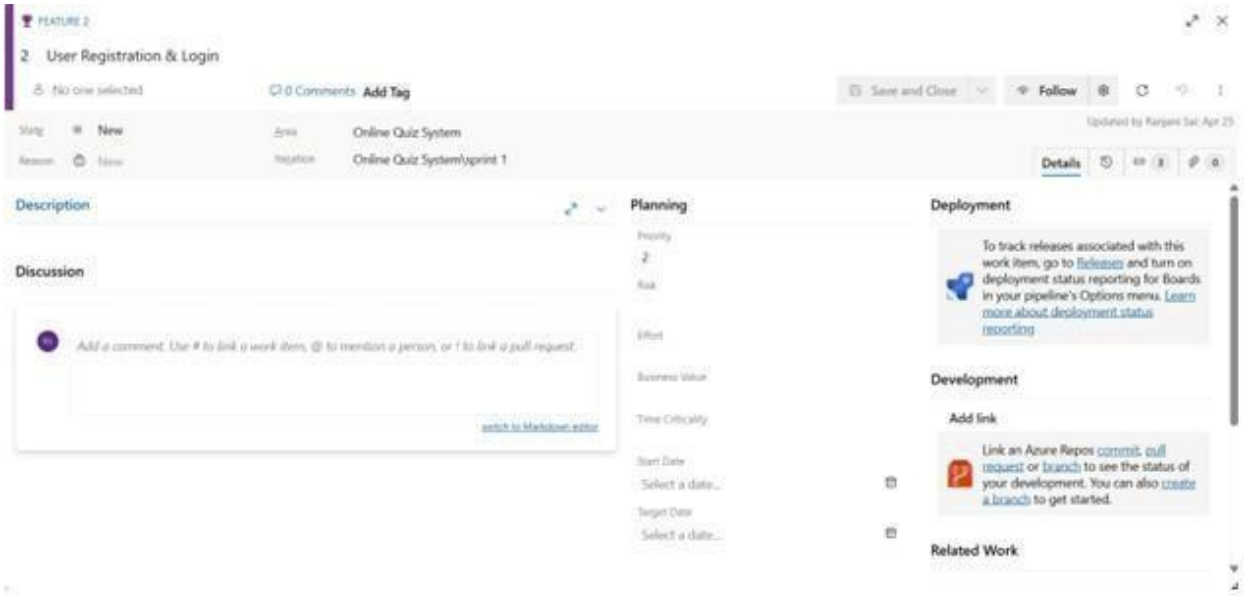
Create Epic, Features, User Stories, Task



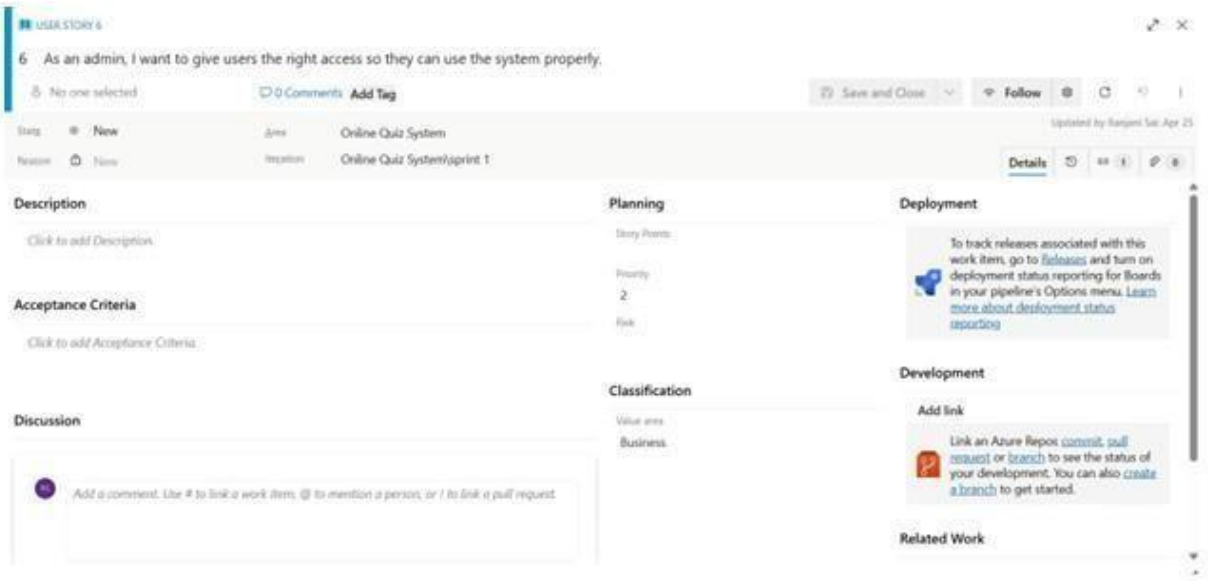
1.Fill in Epics



2.Fill in Features



3.Fill in User Story Details



RESULT :

Thus, the creation of epics, features, user story and task has been created successfully.

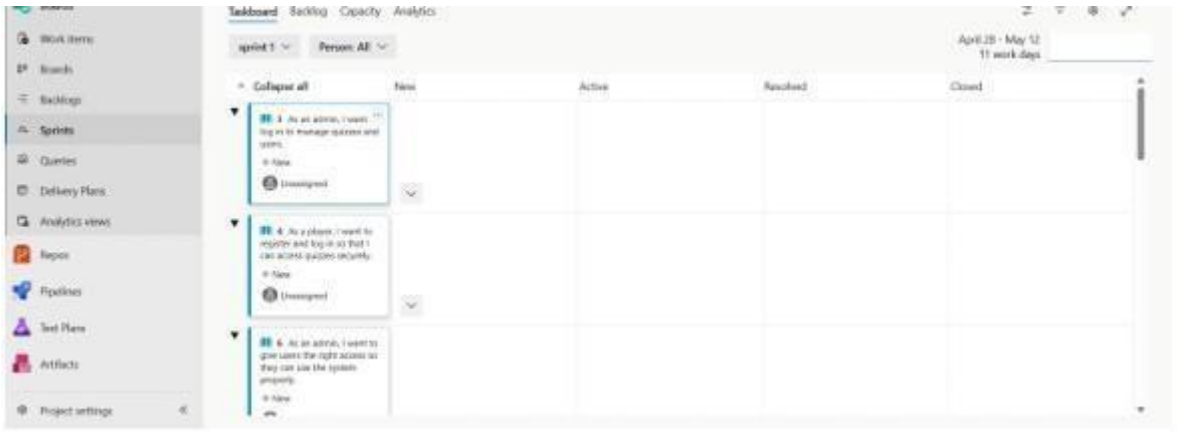
EXP NO:04	**SPRINT PLANNING**

AIM

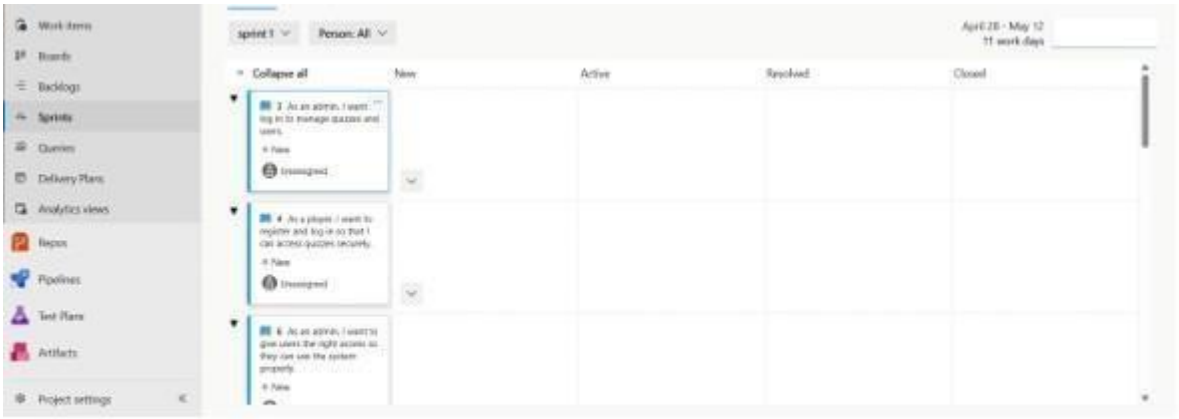
To assign user story to specific sprint for the Online Quiz System**.**

Sprint Planning

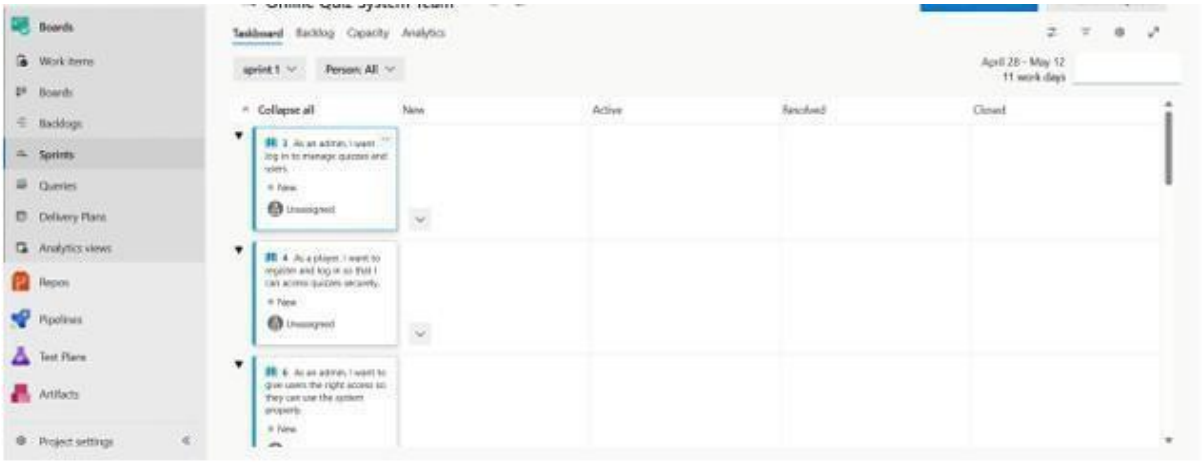
Sprint 1



Sprint 2



Sprint 3



RESULT :

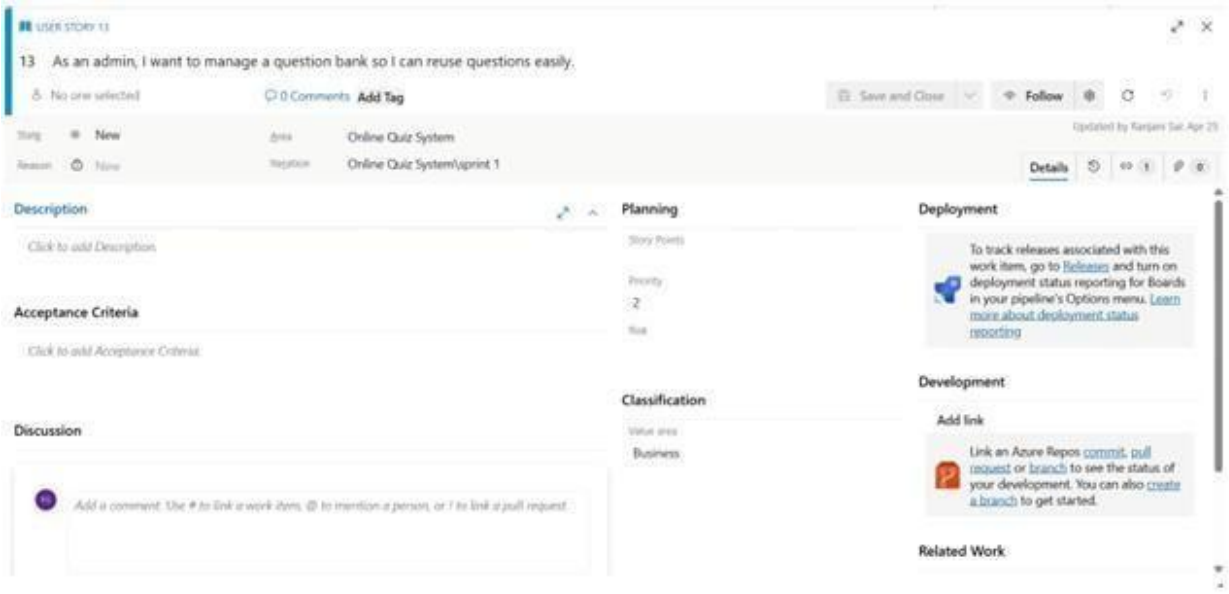
The Sprints have been created for Online Quiz System.

EXP NO :05	POKER ESTIMATION

AIM

Create Poker Estimation for the user stories – Online Quiz System

Poker Estimation



RESULT :

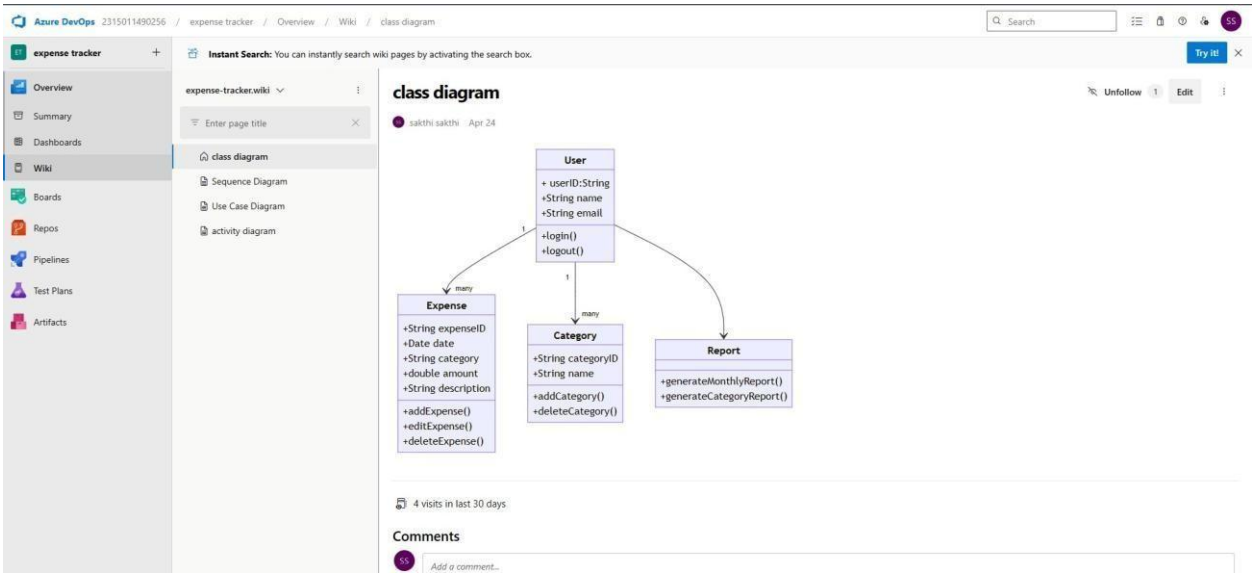
The Estimation/Story Points is created for the project using Poker Estimation.

EXP NO :06	DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE

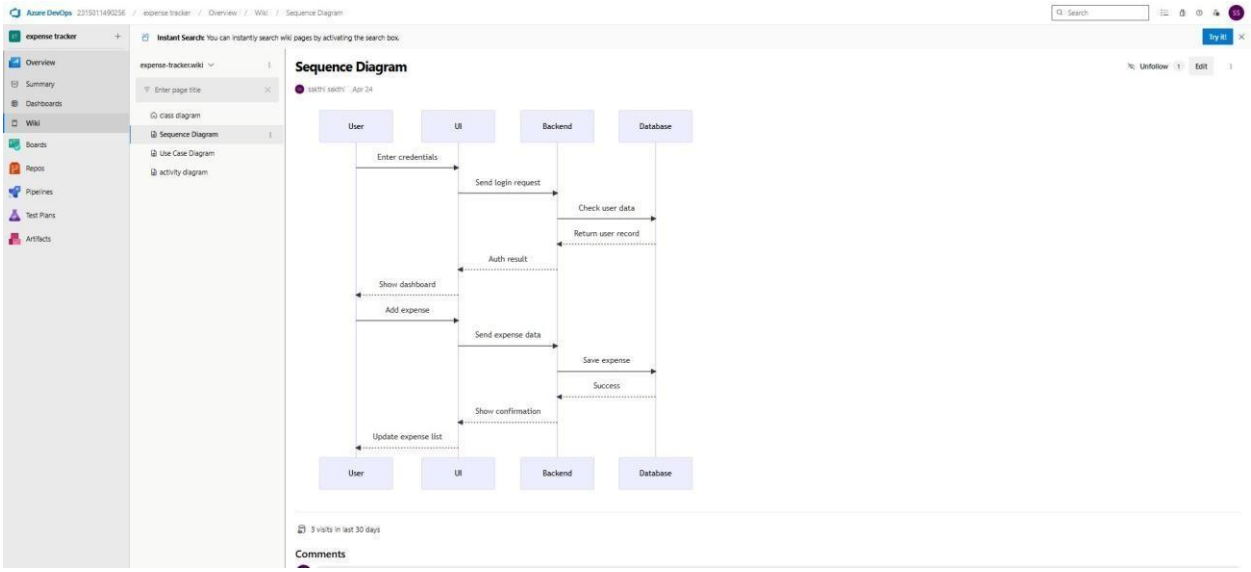
AIM

To Design a Class Diagram and Sequence Diagram for the given Project.

6A. Class Diagram



6B. Sequence Diagram


Azure DevOps
expense tracker
Overview
Summary
Dashboards
Wiki
Boards
Repos
Pipelines
Test Plans
Artifacts
Instant Search: You can instantly search wiki pages by activating the search box.
Try it!
expense-tracker.wiki
Enter page title
class diagram
Sequence Diagram
Use Case Diagram
activity diagram
Sequence Diagram
Unfollow
Edit
User
UI
Backend
Database
Enter credentials
Send login request
Check user data
Return user record
Auth result
Show dashboard
Add expense
Send expense data
Save expense
Success
Show confirmation
Update expense list
User
UI
Backend
Database
3 visits in last 30 days
Comments

RESULT :

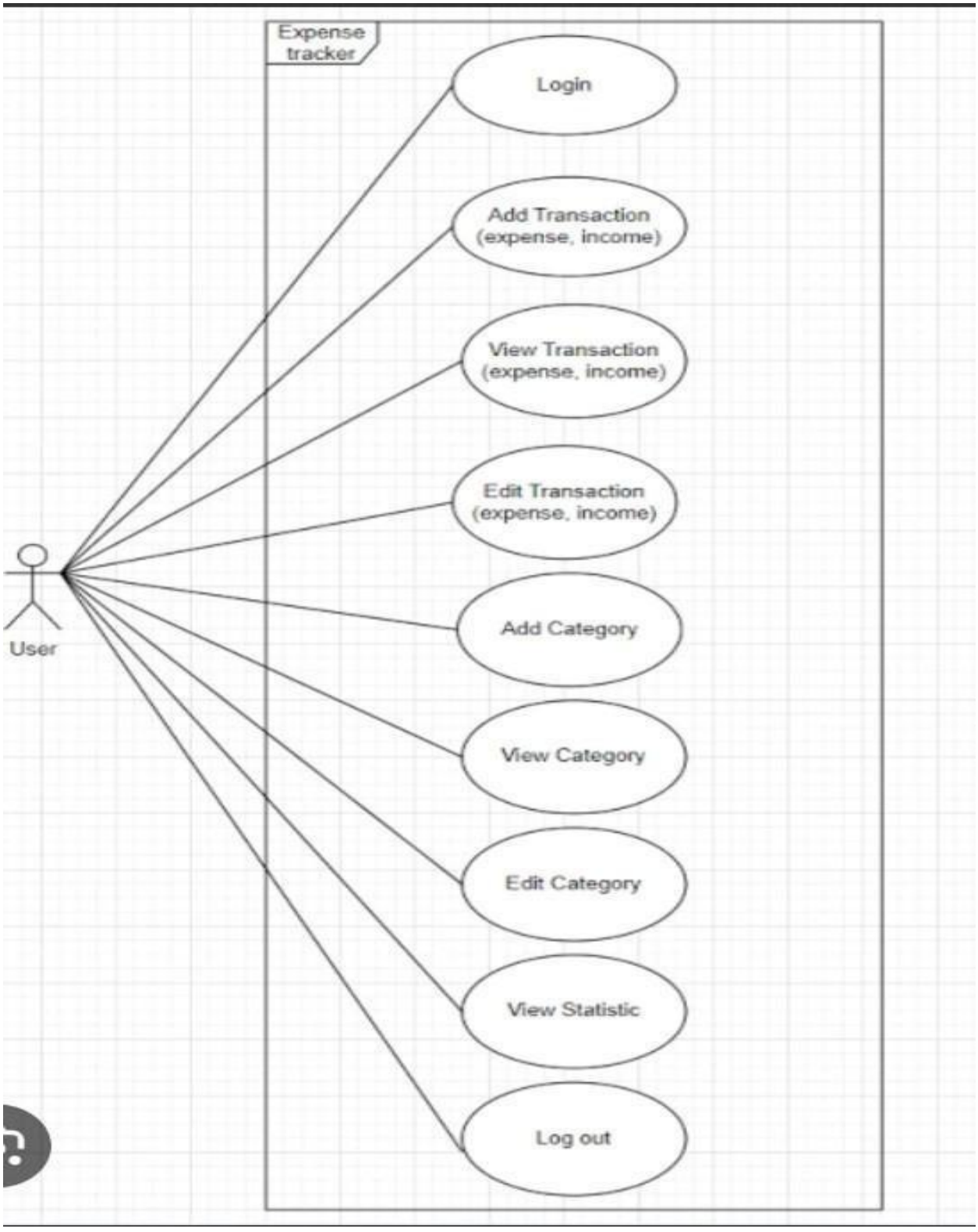
The Class Diagram and Sequence Diagram is designed Successfully for the Online Quiz System.

EXP NO:07	DESIGNING USE-CASE AND ACTIVITY DIAGRAMS FOR PROJECT STRUCTURE

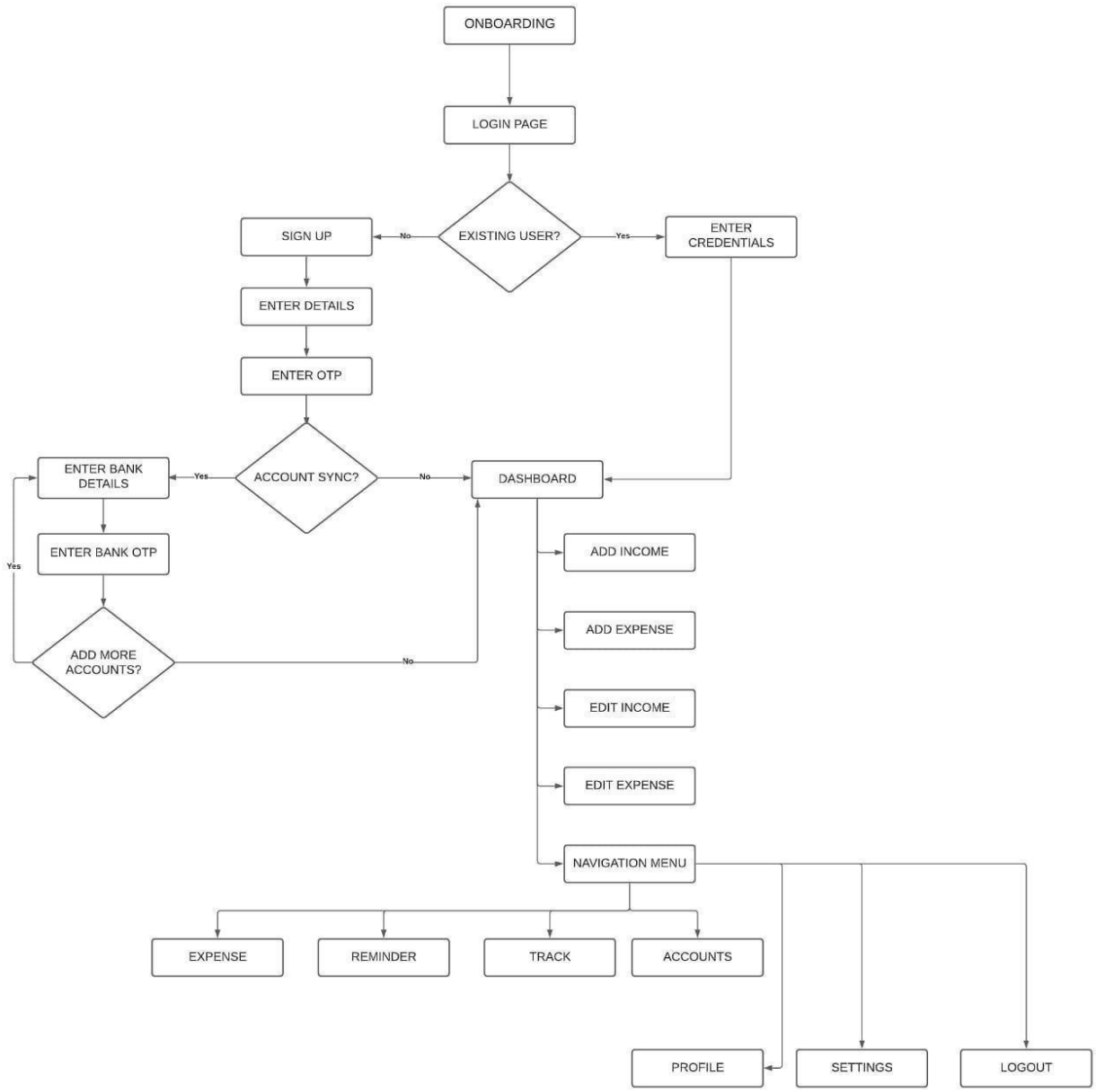
AIM

To Design an Use-Case Diagram and Activity Diagram for the given Project

7A. Use-Case Diagram



7B. Activity Diagram



RESULT :

The Use-Case Diagram and Activity Diagram is designed Successfully for the Online Quiz System.

EXP NO:08	TESTING – TEST PLANS AND TEST CASES

AIM

Test Plans and Test Case and write two test cases for at least five user stories showcasing the happy path and error scenarios in azure DevOps platform.

Test Planning and Test Case Test Case Design Procedure 1. Understand Core Features of the Application

• User Login •

2. Define User Interactions

• Each test case simulates a real user behaviour (e.g., logging in, submitting quizzes, viewing results)

3. Design Happy Path Test Cases

• Focused on validating that all core functionalities work correctly under normal conditions • Example: Player registers and logs in, submits quizzes and views results

4. Design Error Path Test Cases

• Simulate invalid inputs, system issues or failed actions to ensure proper error handling.

• Example: Login with invalid credentials, submission without attachments, unauthorized access to admin panel.

5. Break Down Steps and Expected Results

• Each test case includes a clear sequence of actions and expected results.

• Ensures both manual testers and automation tools can follow the process easily.

6. Use Clear Naming and IDs

• Test cases are uniquely identifies (e.g., TC01 – Valid Login, TC03 – Invalid Password).

• Facilities easy mapping to features and tracking in Azure DevOps.

7. Separate Test

• Grouped by functionality such as:

o Login and Registration

o Quizzes Submission o

Viewing Results o

Admin Functions

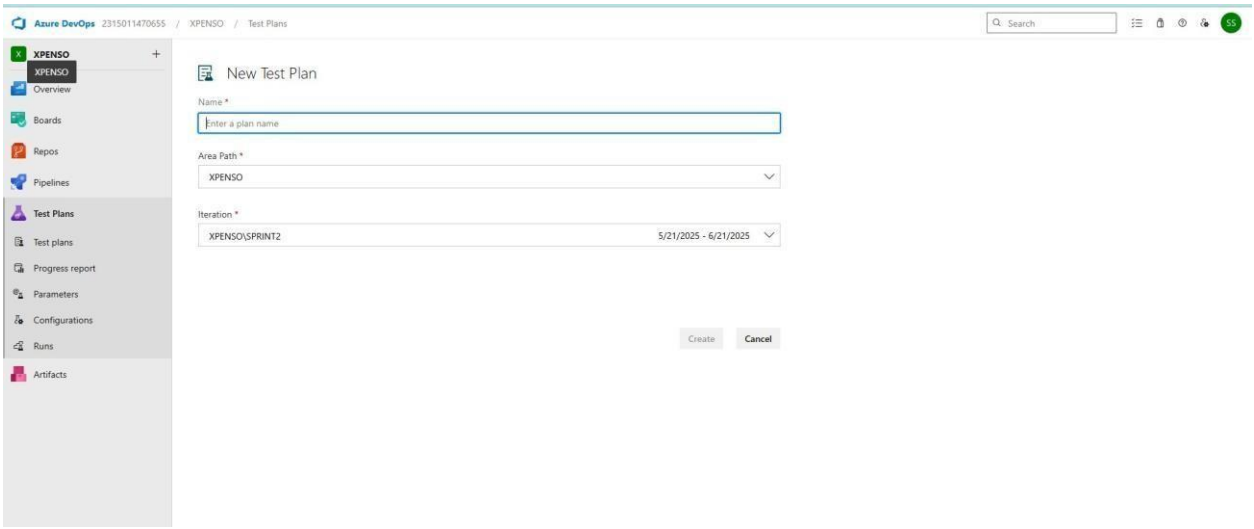
• Improves organization and enables focused execution in Azure DevOps.

8. Prioritize and Review

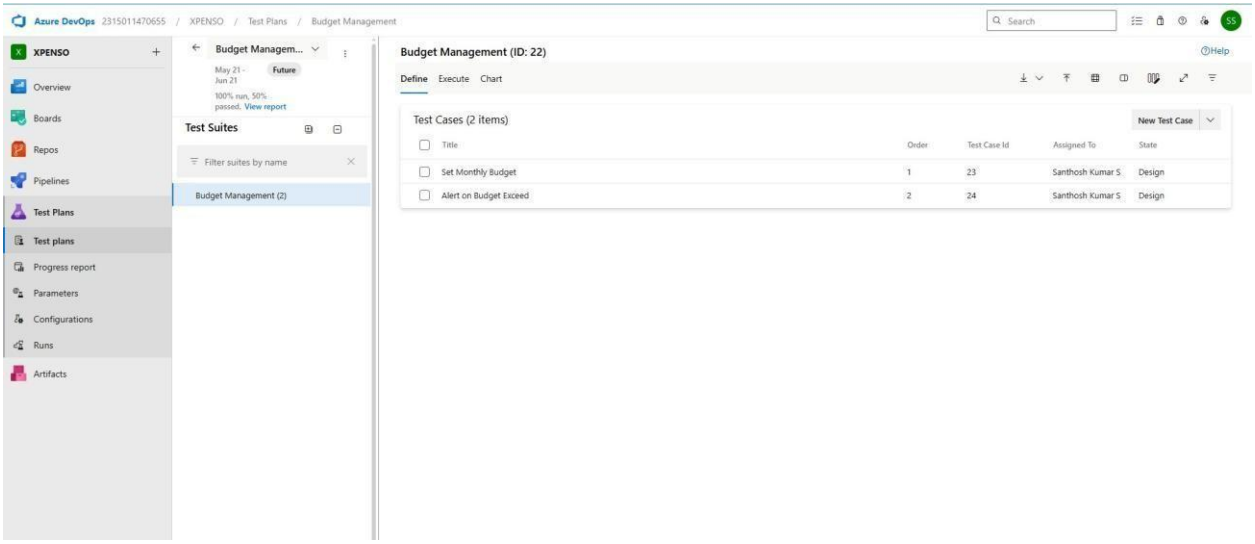
• High-priority assigned to critical workflows like login, quizzes and results.

• Reviewed for compl eteness, accuracy and alignment with user stories and features definition.

1.New test plan



2.Test suite



3.Test case

Give two test cases for at least five user stories showcasing the happy path and error scenarios in azure DevOps platform. Online

Quiz System– Test Plans

USER STORIES

- As an admin, I want to log in to manage quizzes and users (ID: 3).
- As a player, I want to register and log in so that I can access quizzes securely (ID: 4).
- As an admin, I want to give users the right access so they can use the system properly (ID: 6).
- As a player, I want to see only my quizzes and progress so that it's easy to use (ID: 7).
- As an admin, I want to create and configure quizzes with time limits so I can control quiz

TEST SUITES Test Suite: TS01 - Login Verification (ID: 23)

1. TC01 –Valid Login

o Action:

- Go to the Sign-Up page.
- Enter valid name, email, and password.
- Click "Login" button.

o Expected Results:

- Login form is displayed.
- Fields accept valid credentials without error.
- Account is successfully logged in and redirected to the dashboard.

o Type: Happy

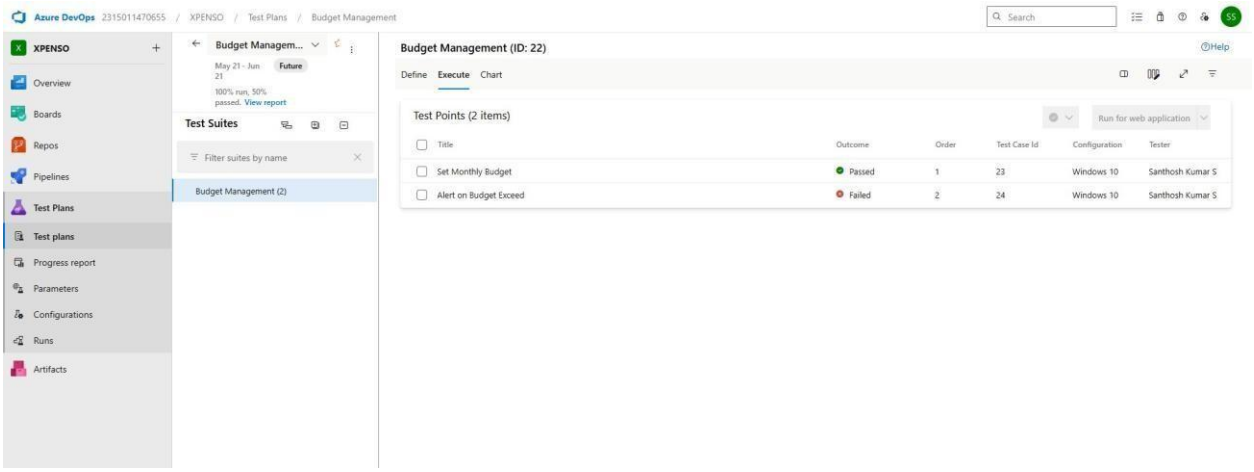
Path 2. TC02 – Invalid Username o

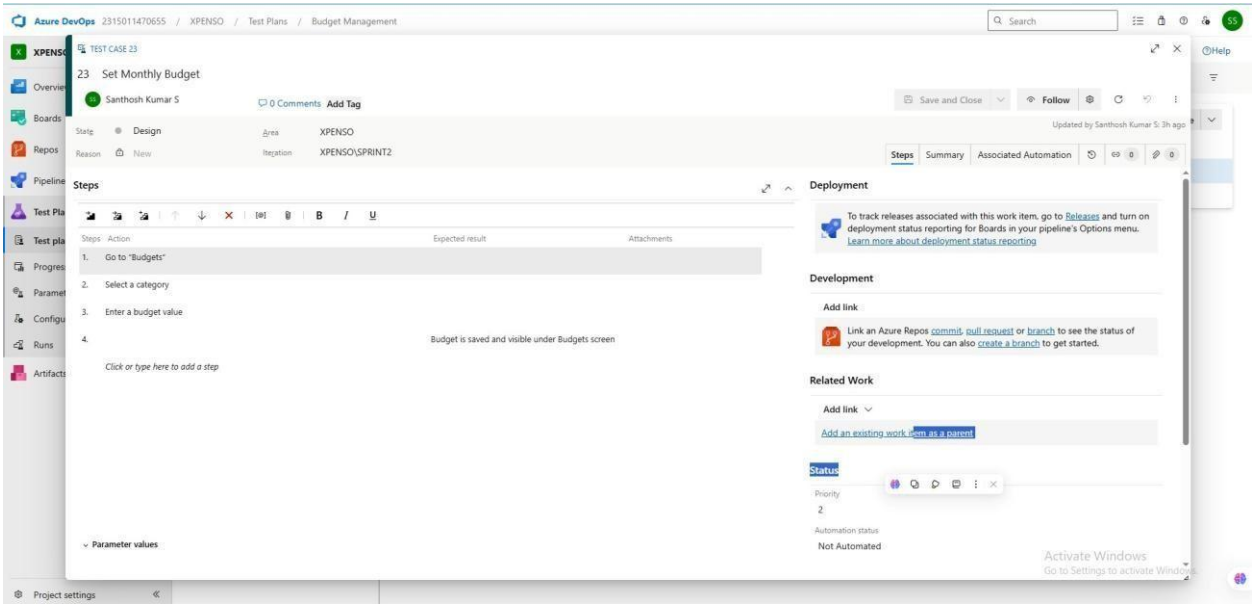
Action:

- Go to the Login page.
- Enter invalid or unregistered username.
- Click on "Login" button. **o**

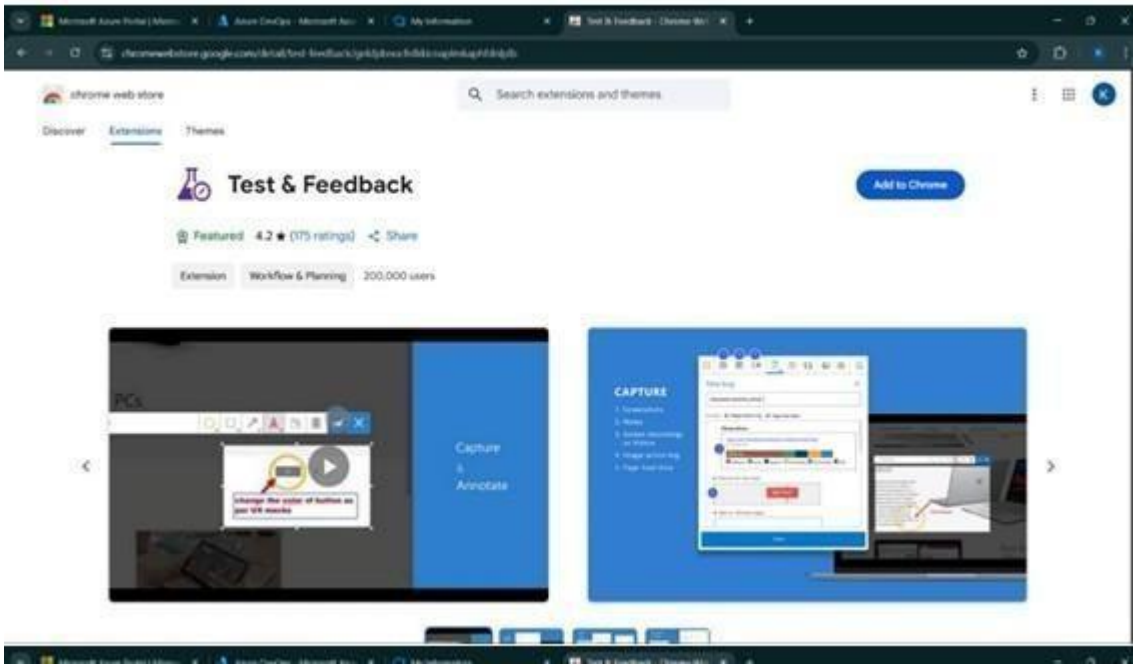
Expected Results:

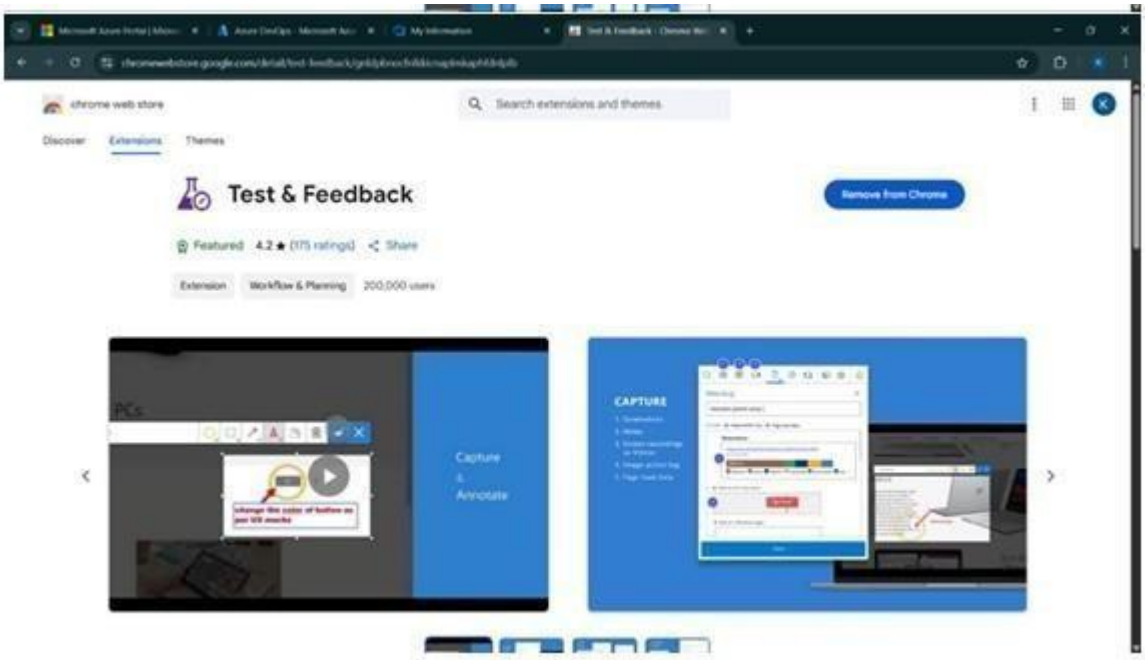
- Login form is displayed.
- System displays an error message like “Invalid username”.
- User is not logged in and remains on the login page.



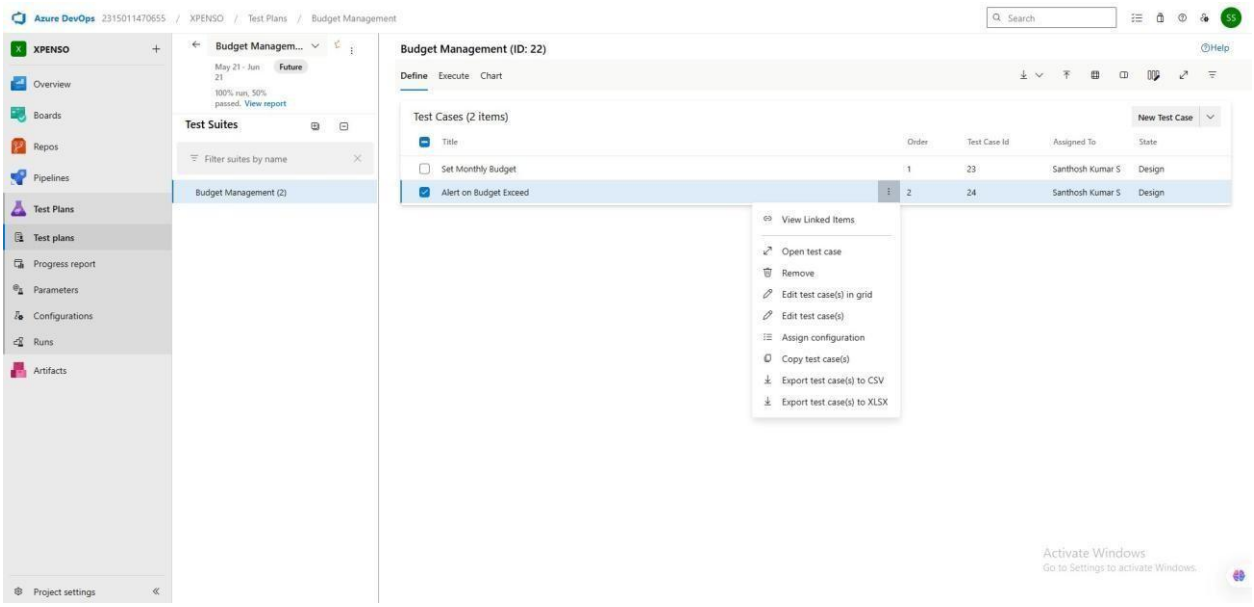


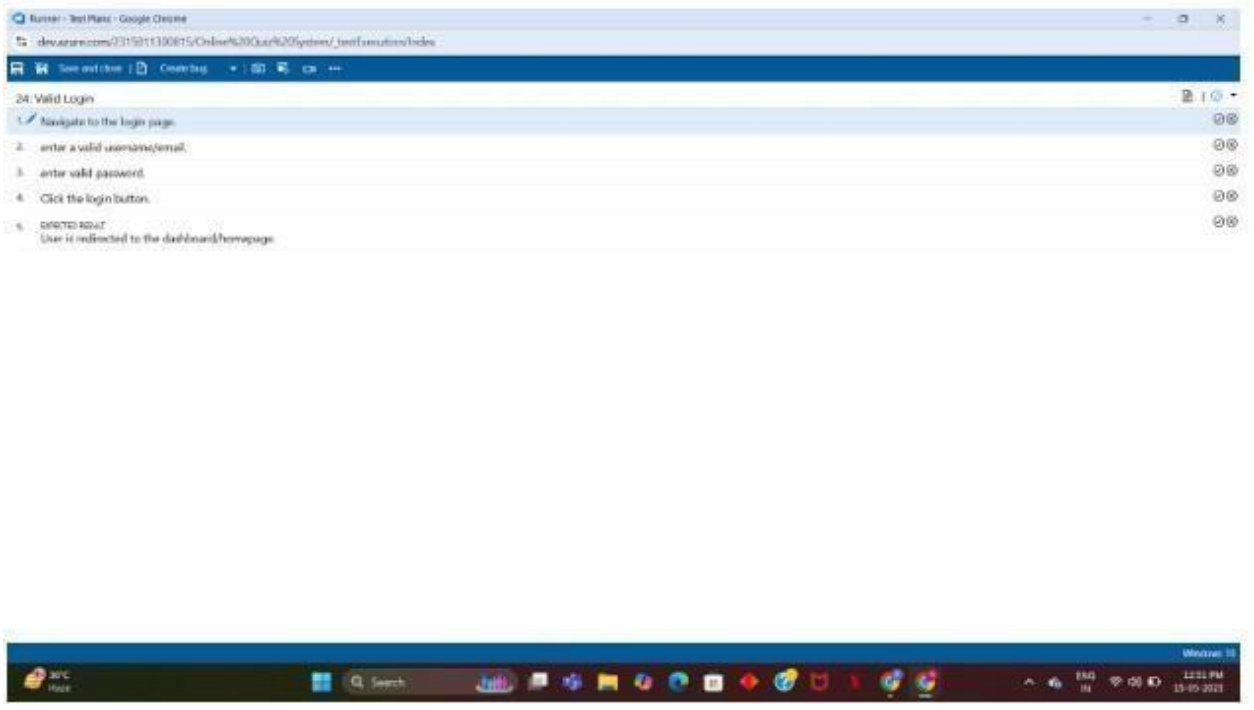
4.Installation of test



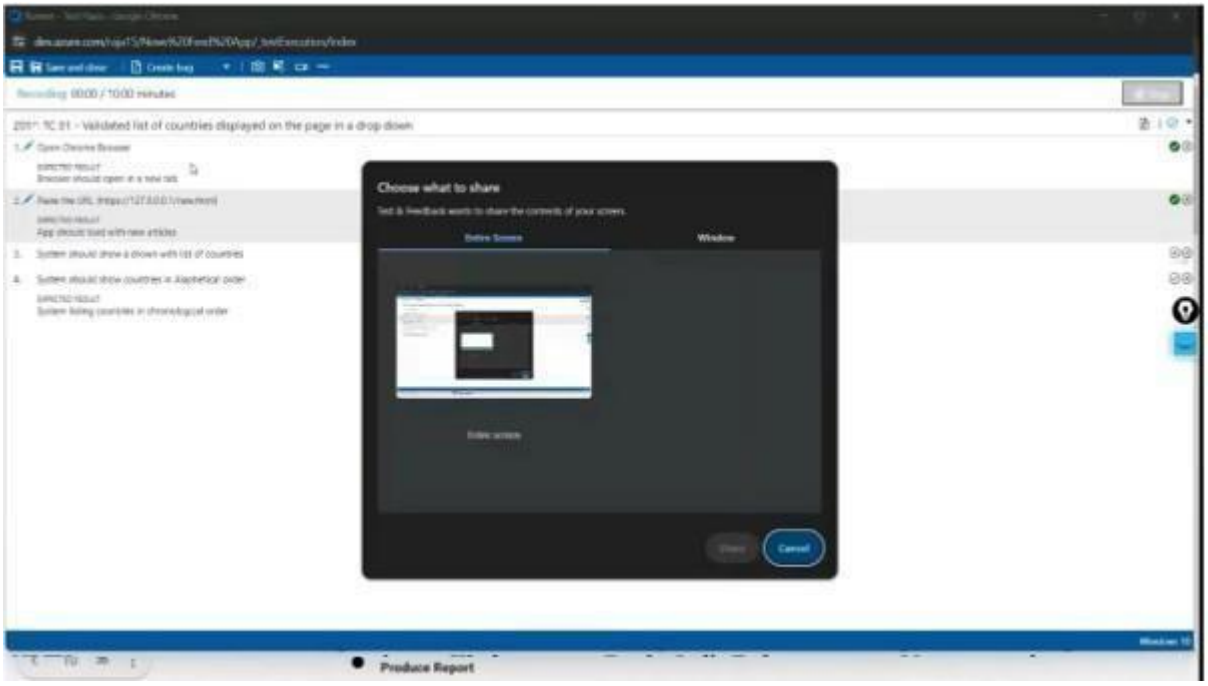


5.Running the test cases

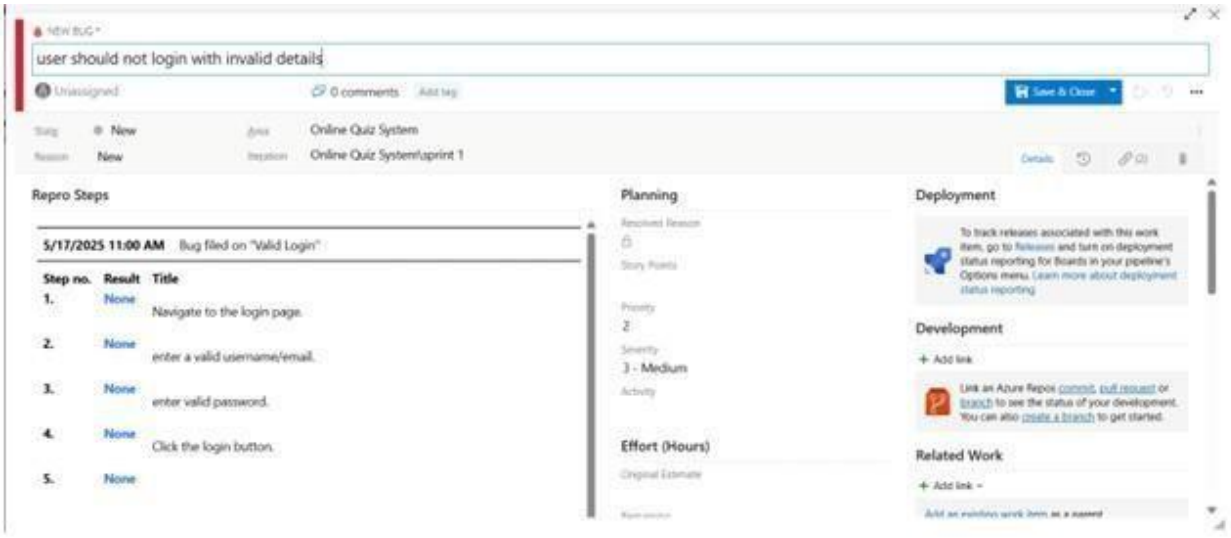


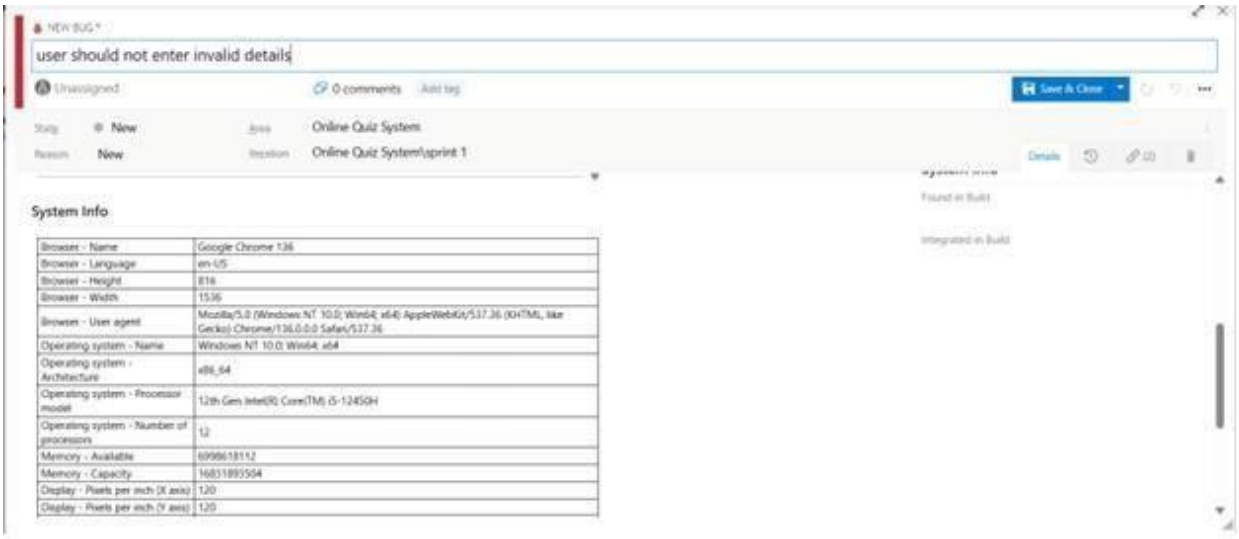


6.Recording the test case

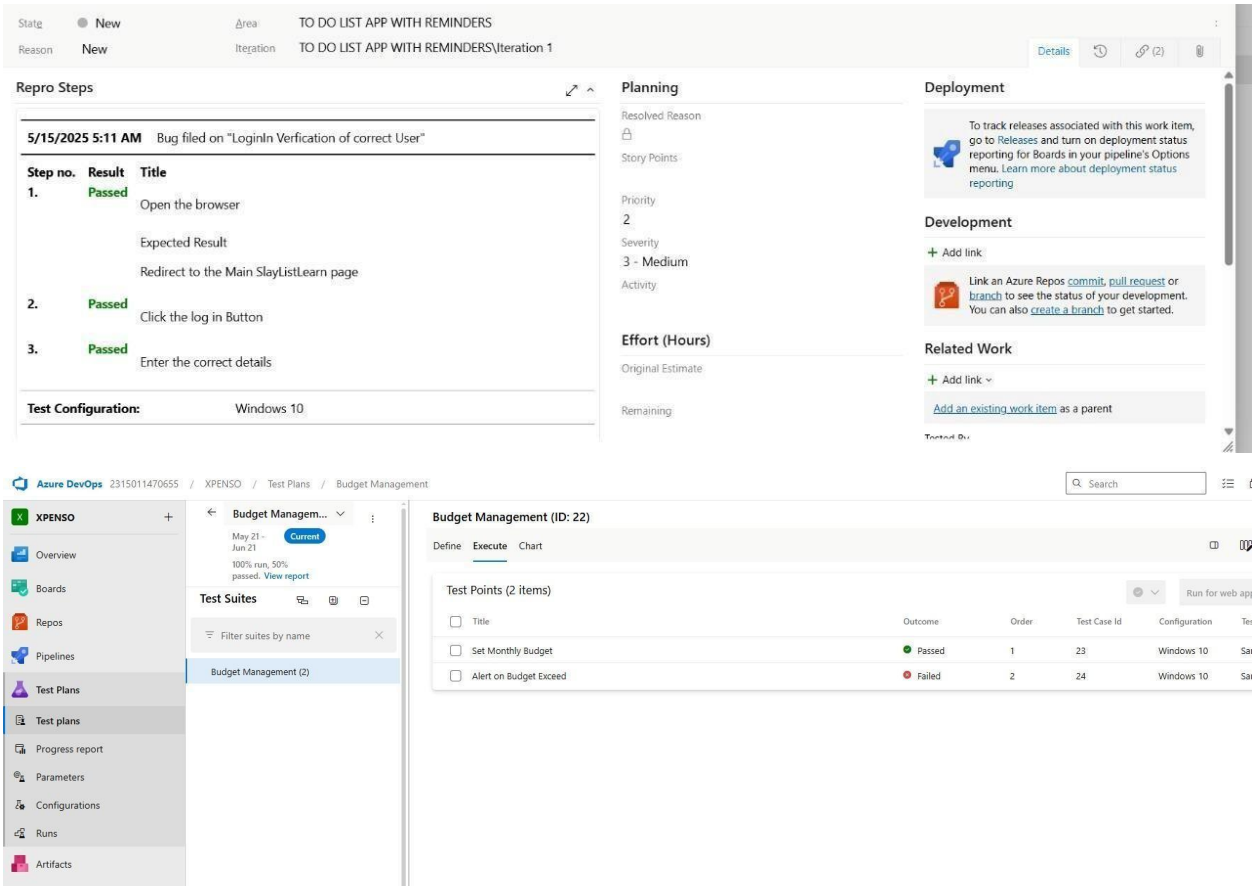


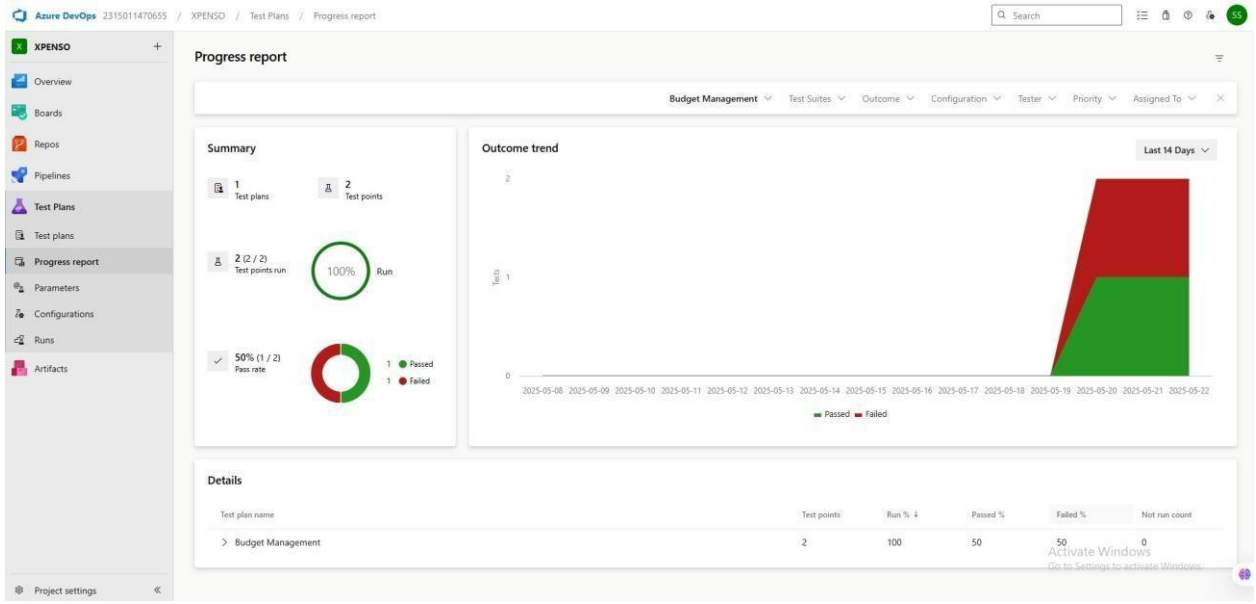
7.Creating the bug



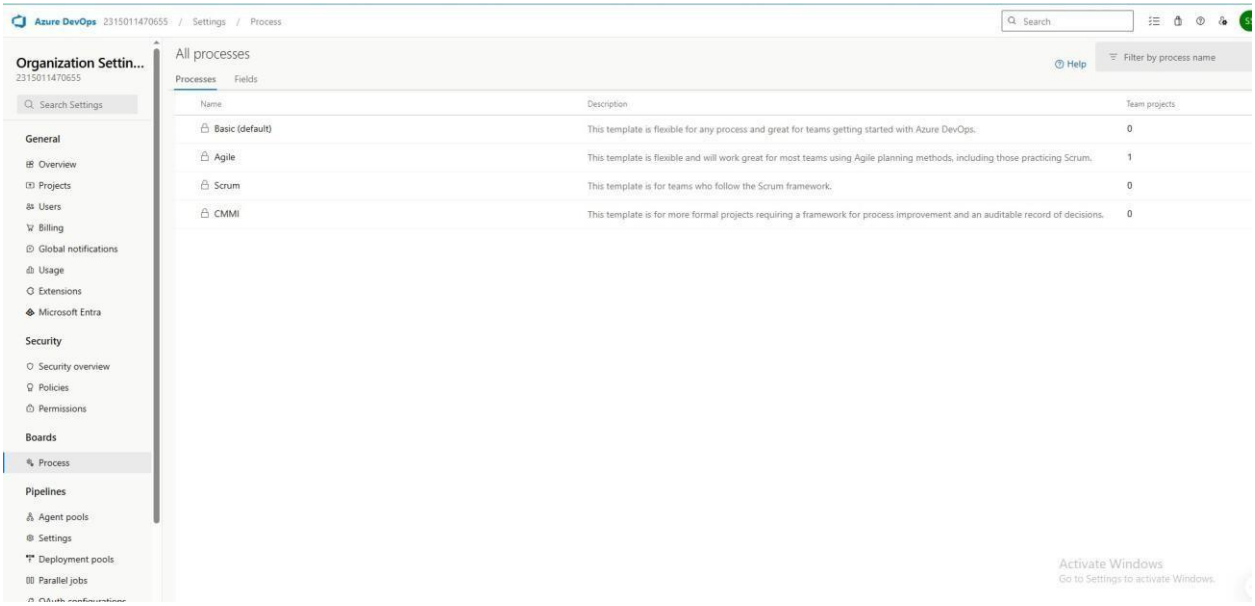


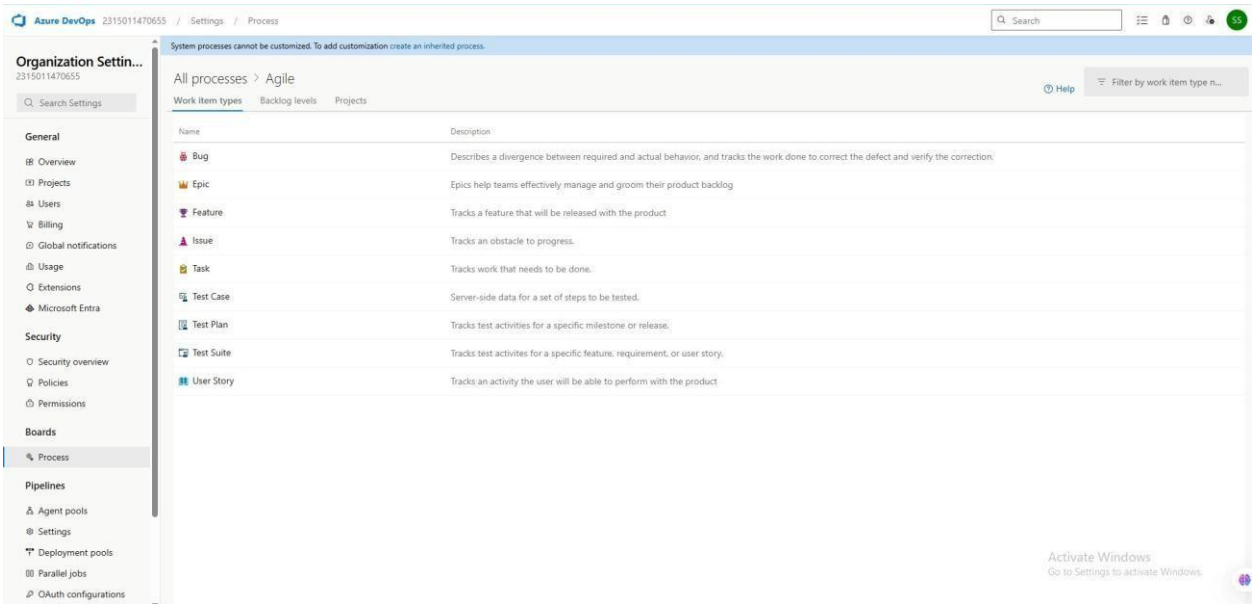
8.Test case results





11.Changing the test template


Azure DevOps 2315011470655 / Settings / Process
Search
Organization Settin...
2315011470655
Search Settings
General
Overview
Projects
Users
Billing
Global notifications
Usage
Extensions
Microsoft Entra
Security
Security overview
Policies
Permissions
Boards
Process
Pipelines
Agent pools
Settings
Deployment pools
Parallel jobs
All processes
Help
Filter by process name
Processes
Fields
Name
Description
Team projects
Basic (default)
This template is flexible for any process and great for teams getting started with Azure DevOps.
0
Agile
This template is flexible and will work great for most teams using Agile planning methods, including those practicing Scrum.
1
Scrum
This template is for teams who follow the Scrum framework.
0
CMMI
This template is for more formal projects requiring a framework for process improvement and an auditable record of decisions.
0
Activate Windows
Go to Settings to activate Windows.


Azure DevOps 2315011470655 / Settings / Process
Search
System processes cannot be customized. To add customization create an inherited process.
Organization Settin...
2315011470655
Search Settings
General
Overview
Projects
Users
Billing
Global notifications
Usage
Extensions
Microsoft Entra
Security
Security overview
Policies
Permissions
Boards
Process
Pipelines
Agent pools
Settings
Deployment pools
Parallel jobs
OAuth configurations
All processes > Agile
Help
Filter by work item type n...
Work item types
Backlog levels
Projects
Name
Description
Bug
Describes a divergence between required and actual behavior, and tracks the work done to correct the defect and verify the correction.
Epic
Epics help teams effectively manage and groom their product backlog
Feature
Tracks a feature that will be released with the product
Issue
Tracks an obstacle to progress.
Task
Tracks work that needs to be done.
Test Case
Server-side data for a set of steps to be tested.
Test Plan
Tracks test activities for a specific milestone or release.
Test Suite
Tracks test activites for a specific feature, requirement, or user story.
User Story
Tracks an activity the user will be able to perform with the product
Activate Windows
Go to Settings to activate Windows.

RESULT :

The test plans and test cases for the user stories is created in Azure DevOps with Happy Path and Error Path.

EXP NO :09	**CI/CD PIPELINES IN AZURE**

AIM

To implement a Continuous Integration and Continuous Deployment (CI/CD) pipeline in Azure DevOps for automating the build, testing, and deployment process of the Student Management System, ensuring faster delivery and improved software quality.

PROCEDURE

Steps to Create and implement pipelines in Azure:

1. Sign in to Azure DevOps and Navigate to Your Project Log in to dev.azure.com, select your organization, and open the project where your Student Management System code resides.

2. Connect a Code Repository (Azure Repos or GitHub) Ensure your application code is stored in a Gitbased repository such as Azure Repos or GitHub. This will be the source for triggering builds and deployments in your pipeline.

3. Create a New Pipeline Go to the Pipelines section on the left panel and click “Create Pipeline”. Choose your source (e.g., Azure Repos Git or GitHub), and then select the repository containing your project code.

4. Choose the Pipeline Configuration You can select either the YAML-based pipeline (recommended for version control and automation) or the Classic Editor for a GUI-based setup. If using YAML, Azure

   DevOps will suggest a template or allow you to define your own.

5. Define Build Stage (CI - Continuous Integration) from YAML file.

6. Install dependencies (e.g., npm install, dotnet restore).

7. Build the application (dotnet build, npm run build).

8. Run unit tests (dotnet test, npm test).

9. Publish build artifacts to be used in the release stage.

10. Save and Run the Pipeline for the First Time Save the YAML or build definition and click “Run”.

    Azure will fetch the latest code and execute the defined build and test stages.

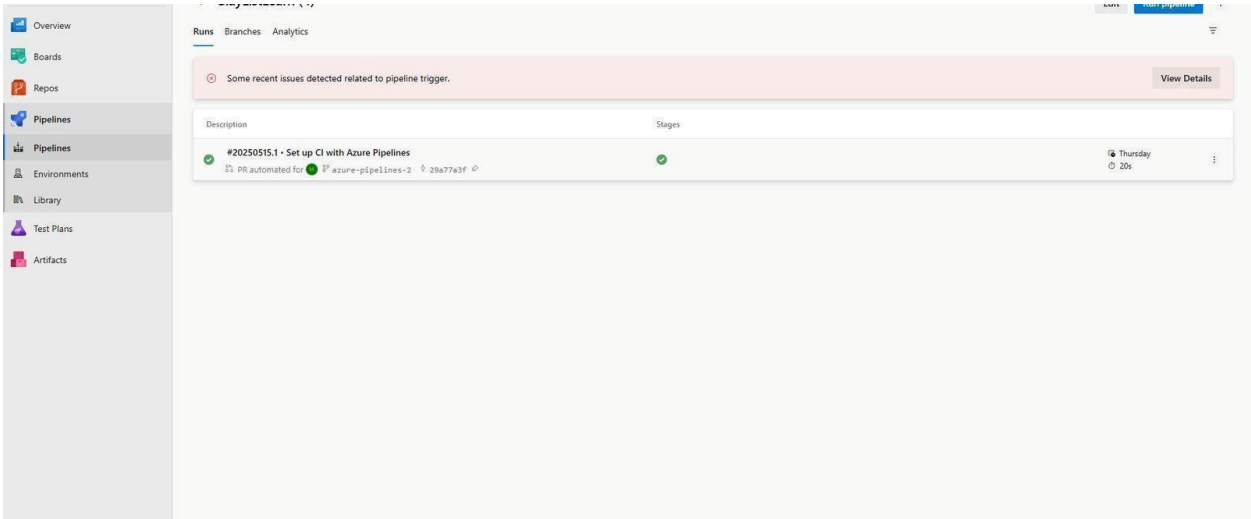
11. Configure Continuous Deployment (CD) Navigate to the Releases tab under Pipelines and click “New Release Pipeline”. Add an Artifact (from the build stage) and create a new Stage (e.g., Development, Production).

12. Configure the CD stage with deployment tasks such as deploying to Azure App Service, running database migrations or scripts, and restarting services using the Azure App Service Deploy task linked to your subscription and app details.

13. Set Triggers and Approvals Enable continuous deployment trigger so the release pipeline runs automatically after a successful build. For production environments, configure pre-deployment approvals to ensure manual verification before release.

14. Monitor Pipelines and Manage Logs View all pipeline runs under the Runs section. Check logs for build/test/deploy stages to debug any errors. You can also integrate email alerts or Microsoft Teams notifications for build failures.

15. Review and Maintain Pipelines Regularly update your pipeline tasks or YAML configurations as your application grows. Ensure pipeline runs are clean and artifacts are stored securely. Integrate quality gates and code coverage policies to maintain code quality.



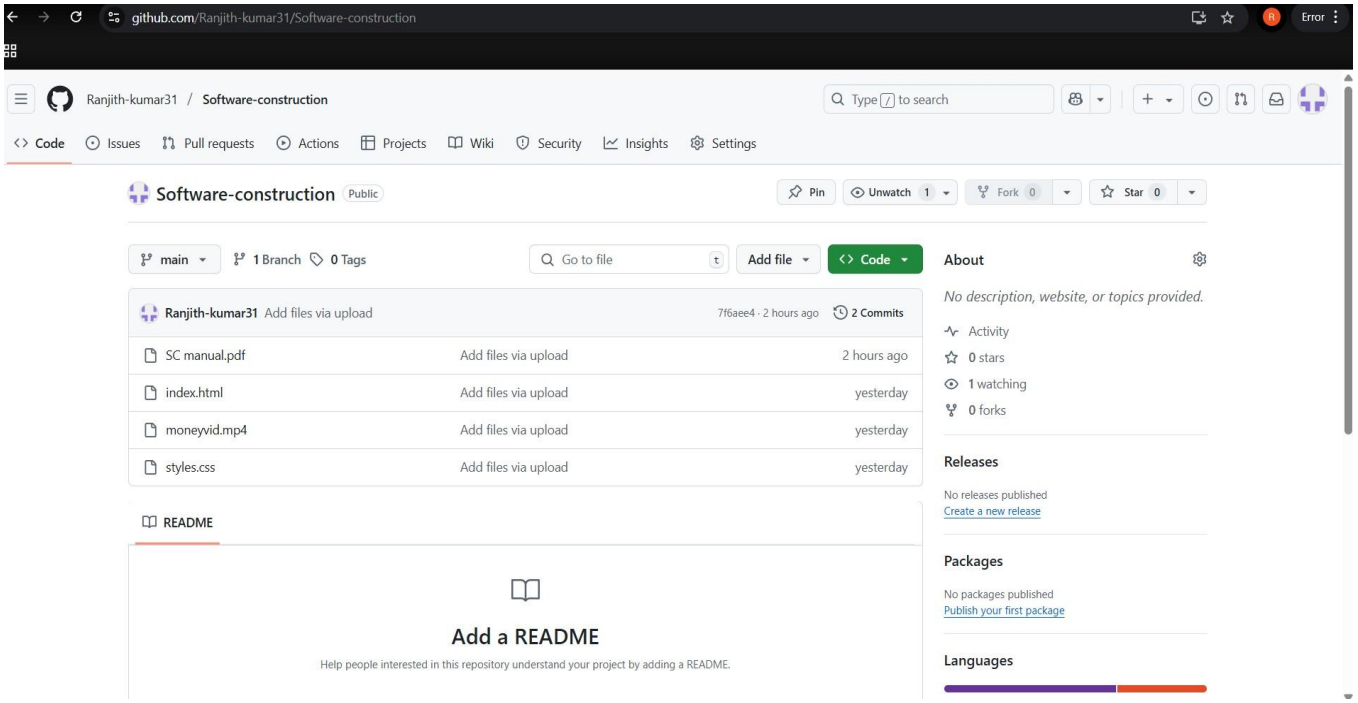
RESULT : Thus the pipelines for the given project “Online Quiz System” has been executed successfully.

EXP NO: 10	GITHUB: PROJECT STRUCTURE & NAMING CONVENTIONS

Aim:

To provide a clear and organized view of the project's folder structure and file naming conventions, helping contributors and users easily understand, navigate, and extend the Online Quiz System.

Github:



RESULT :

The GitHub repository clearly displays the organized project structure and consistent naming conventions, making it easy for users and contributors to understand and navigate the codebase.